

Python: module vcs.outline

vcs.outline

[index](#)

Outline ([Go](#)) module

Modules

[vcs.Canvas](#)

[vcs.VCS validation functions](#)

[vcs. vcs](#)

[cdtime](#)

[vcs.queries](#)

[vcs](#)

Classes

[builtin .object](#)

[Go](#)

class **Go**([builtin .object](#))

Class: [Go](#)

Outline

Description of [Go](#) Class:

The Outline graphics method outlines a set of integer values in a type array that indicates land, ocean, and sea-ice points. The example shows how to change such an outline by modifying its attributes.

Other Useful Functions:

a=vcs.init()	# Constructor
a.show('outline')	# Show predefined outline gr...
a.show('line')	# Show predefined VCS line ol...
a.setcolormap("AMIP")	# Change the VCS color map
a.outline(s,o,'default')	# Plot data 's' with outline 'default' template
a.update()	# Updates the VCS Canvas at ...
a.mode=1, or 0	# If 1, then automatic update; 0, then use update function to update the VCS Canvas.

Example of Use:

a=vcs.init()

To Create a new instance of outline use:

out=a.createoutline('new','quick') # Copies content of 'quick'
out=a.createoutline('new') # Copies content of 'default'

To Modify an existing outline use:

```

out=a.getoutline('AMIP_psl')

out.list()                                # Will list all the outline
out.projection='linear'
lon30={-180:'180W',-150:'150W',0:'Eq'}
out.xticlabels1=lon30
out.xticlabels2=lon30
out.xticlabels(lon30, lon30)               # Will set them both
out.xmtics1=''
out.xmtics2=''
out.xmtics(lon30, lon30)                 # Will set them both
out.yticlabels1=lat10
out.yticlabels2=lat10
out.yticlabels(lat10, lat10)               # Will set them both
out.ymtics1=''
out.ymtics2=''
out.ymtics(lat10, lat10)                 # Will set them both
out.datawc_y1=-90.0
out.datawc_y2=90.0
out.datawc_x1=-180.0
out.datawc_x2=180.0
out.datawc(-90, 90, -180, 180)           # Will set them all
xaxisconvert='linear'
yaxisconvert='linear'
out.xyscale('linear', 'area_wt')          # Will set them both

Specify the outline fill values:
out.outline=([0,1,2,3,4])                # Same as below
out.outline=(0,1,2,3,4)                   # Will specify the outline va

Specify the outline line type:
out.line=0                                 # same as out.line = 'solid'
out.line=1                                 # same as out.line = 'dash'
out.line=2                                 # same as out.line = 'dot'
out.line=3                                 # same as out.line = 'dash-dot'
out.line=4                                 # same as out.line = 'long-dash'

There are four possibilities for setting the line color index (Ex):
out.linecolor=22                           # Same as below
out.linecolor=(22)                         # Same as below
out.linecolor=([22])                       # Will set the outline to a
                                           # color index
out.linecolor=None                         # Turns off the color index

out.linewidth=1                            # width range: 1 to 100, def

```

Methods defined here:

[__init__](#)(self, parent, Go_name=None, Go_name_src='default', createGo=0)

[datawc](#)(self, dsp1=1e+20, dsp2=1e+20, dsp3=1e+20, dsp4=1e+20)

```

list(self)

rename = renameGo(self, old_name, new_name)
#####
#
# Function:      renameGo
#
# Description of Function:
#      Private function that renames the name of an existing
#      graphics method.
#
#
# Example of Use:
#      renameGo(old_name, new_name)
#              where: old_name is the current name of outline
#                      new_name is the new name for the outline
#
#####
script(self, script_filename=None, mode=None)
    Function:      script                                # Calls _vcs.s

    Description of Function:
        Saves out a outline graphics method in Python or VCS sc
        designated file.

    Example of Use:
        script(scriptfile_name)
            where: script_name is the output name of the scr
                    mode is either "w" for replace or "a" for

        Note: If the the filename has a ".py" at the end
              Python script. If the filename has a ".scr"
              produce a VCS script. If neither extension
              default a Python script will be produced.

    a=vcs.init()
    out=a.createboxfill('temp')
    out.script('filename.py')                         # Append to a Python fil
    out.script('filename.scr')                       # Append to a VCS file "
    out.script('filename', 'w')

xmtics(self, xmt1=", xmt2=")

xticlabels(self, xtl1=", xtl2=")

xyscale(self, xat=", yat=")

ymtics(self, ymt1=", ymt2=")

yticlabels(self, ytl1=", ytl2=")

```

Properties defined here:

```
datawc_calendar
    get">get = _getcalendar(self)
    set">set = _setcalendar(self, value)

datawc_timeunits
    get">get = _gettimeunits(self)
    set">set = _settimeunits(self, value)

datawc_x1
    get">get = _getdatawc_x1(self)
    set">set = _setdatawc_x1(self, value)

datawc_x2
    get">get = _getdatawc_x2(self)
    set">set = _setdatawc_x2(self, value)

datawc_y1
    get">get = _getdatawc_y1(self)
    set">set = _setdatawc_y1(self, value)

datawc_y2
    get">get = _getdatawc_y2(self)
    set">set = _setdatawc_y2(self, value)

line
    get">get = _getline(self)
    set">set = _setline(self, value)

linecolor
    get">get = _getlinecolor(self)
    set">set = _setlinecolor(self, value)

linewidth
    get">get = _getlinewidth(self)
    set">set = _setlinewidth(self, value)

name
    get">get = _getname(self)
    set">set = _setname(self, value)

outline
    get">get = _getoutline(self)
    set">set = _setoutline(self, value)

projection
    get">get = _getprojection(self)
    set">set = _setprojection(self, value)

xaxisconvert
    get">get = _getxaxisconvert(self)
```

```
set">set = _setxaxisconvert(self, value)
```

xmtics1

```
get">get = _getxmtics1(self)  
set">set = _setxmtics1(self, value)
```

xmtics2

```
get">get = _getxmtics2(self)  
set">set = _setxmtics2(self, value)
```

xticlabels1

```
get">get = _getxticlabels1(self)  
set">set = _setxticlabels1(self, value)
```

xticlabels2

```
get">get = _getxticlabels2(self)  
set">set = _setxticlabels2(self, value)
```

yaxisconvert

```
get">get = _getyaxisconvert(self)  
set">set = _setyaxisconvert(self, value)
```

ymtics1

```
get">get = _getymtics1(self)  
set">set = _setymtics1(self, value)
```

ymtics2

```
get">get = _getymtics2(self)  
set">set = _setymtics2(self, value)
```

yticlabels1

```
get">get = _getyticlabels1(self)  
set">set = _setyticlabels1(self, value)
```

yticlabels2

```
get">get = _getyticlabels2(self)  
set">set = _setyticlabels2(self, value)
```

Data and other attributes defined here:

__slots__ = ['setmember', 'parent', 'name', 'g_name', 'xaxisconvert', 'yaxisconvert', 'outline', 'linecolor', 'projection', 'xticlabels1', 'xticlabels2', 'yticlabels1', 'yticlabels2', 'xmtics1', 'xmtics2', 'ymtics1', 'ymtics2']

g_name = <member 'g_name' of 'Go' objects>

parent = <member 'parent' of 'Go' objects>

setmember = <member 'setmember' of 'Go' objects>

Functions

```

getGomember(self, member)
#####
#
# Function:      getGomember
#
# Description of Function:
#      Private function that retrieves the outline members from t
#      structure and passes it back to Python.
#
#
# Example of Use:
#      return_value =
#      getGomember(self, name)
#              where: self is the class (e.g., Go)
#                      name is the name of the member that is being
#
#####
#
getmember = getGomember(self, member)
#####
#
# Function:      getGomember
#
# Description of Function:
#      Private function that retrieves the outline members from t
#      structure and passes it back to Python.
#
#
# Example of Use:
#      return_value =
#      getGomember(self, name)
#              where: self is the class (e.g., Go)
#                      name is the name of the member that is being
#
#####
#
renameGo(self, old_name, new_name)
#####
#
# Function:      renameGo
#
# Description of Function:
#      Private function that renames the name of an existing outl
#      graphics method.
#
#
# Example of Use:
#      renameGo(old_name, new_name)
#              where: old_name is the current name of outline gra
#                      new_name is the new name for the outline gra
#
#####
#

```

```

setGomember(self, member, value)
#####
#
# Function:      setGomember
#
# Description of Function:
#      Private function to update the VCS canvas plot. If the can
#      set to 0, then this function does nothing.
#
#
# Example of Use:
#      setGomember(self,name,value)
#          where: self is the class (e.g., Go)
#                  name is the name of the member that is being
#                  value is the new value of the member (or att
#
#####
setmember = setGomember(self, member, value)
#####
#
# Function:      setGomember
#
# Description of Function:
#      Private function to update the VCS canvas plot. If the can
#      set to 0, then this function does nothing.
#
#
# Example of Use:
#      setGomember(self,name,value)
#          where: self is the class (e.g., Go)
#                  name is the name of the member that is being
#                  value is the new value of the member (or att
#
#####

```

Data

StringTypes = (<type 'str'>, <type 'unicode'>)